Perencanaan Jadwal Dengan *Graph Colouring*

Ferry Mulia
NIM 13506047

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha no. 10 Bandung
E-mail : if16047@students.if.itb.ac.id

Abstrak

Makalah ini membahas mengenai metode-metode mengenai cara membuat/merencanakan jadwal dengan menggunakan teknik mewarnai graf atau *graph colouring*.Pewarnaan graf di sini adalah pewarnaan simpul atau *vertex colouring*.

Pewarnaan simpul adalah teknik mewarnai simpul-simpul pada graf sehingga tidak ada simpul-simpul yang bertetangga, yaitu terhubung langsung dengan minimal sebuah sisi, memiliki warna yang sama. Biasanya hal ini juga dikaitkan dengan penggunaan warna yang seminimal mungkin.

Teknik pewarnaan graf merupakan salah satu subjek yang menarik dan terkenal dalam bidang graf. Teori-teori mengenainya telah banyak dikembangkan dan berbagai algoritma dengan kelebihan dan kelemahan masing-masing telah dibuat untuk menyelesaikannya. Aplikasi dari teknik ini juga telah banyak diterapkan di berbagai bidang, salah satunya adalah membuat jadwal. Perencanaan jadwal di sini khususnya diterapkan pada pekerjaan-pekerjaan atau hal-hal yang saling terkait, misalnya hal-hal yang berlangsung pada waktu sama, atau pekerjaan yang menggunakan sumber daya sama, dan sebagainya. Teknik pewarnaan graf akan membuat jadwal kerja yang dapat menghasilkan hasil yang meksimum dengan cara yang paling efisien.

Kata Kunci : graf, *graph colouring*, bilangan kromatik, *scheduling optimization*, *graph colouring algorithm*, aplikasi pewarnaan graph.

1. Pendahuluan

Graf adalah salah satu pokok bahasan Matematika Diskrit yang telah lama dikenal dan banyak diaplikasikan pada berbagai bidang. Secara umum, graf adalah pasangan himpunana (V,E) dimana V adalah himpunana tidak kosong dari simpul-simpul (*vertex* atau *node*) dan E adalah himpunana sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul pada graf tersebut.

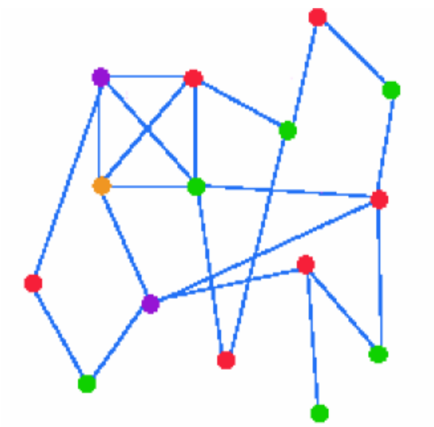
$V = \{v_1, v_2, v_3, \ldots , v_n\}$
$E = \{e_1, e_2, e_3, \ldots , e_n\}$
Atau
$E=\{(v_1 , v_2),(v_2 , v_3),(v_3 , v_4), \ldots ,(v_{n-1} , v_n)\}$
Dimana $e=(v_i,v_j)$ yang artinya sisi yang menghubungkan simpul v_i dan v_j. [6]

Kegunaan graf sangat banyak. Umumnya graf digunakan untuk memodelkan suatu masalah sehingga menjadi lebih mudah, yaitu dengan cara merepresentasikan objek-objek tersebut. Contoh pemodelan suatu masalah dengan menggunakan graf dapat dilihat pada penggambaran rangkaian listrik, senyawa kimia, jaringan komunikasi, jaringan network komputer, analisis algoritma, peta, struktur hierarki sosial, dan lain-lain.

Salah satu topik yang menarik adalah masalah pewarnaan graf (*graph colouring*). Bidang ini memiliki sejarah menarik dan teori-teorinya telah menimbulkan banyak perdebatan pada kalangan matematikawan.

Terdapat tiga macam pewarnaan graf yaitu pewarnaan simpul (vertex colouring), pewarnaan sisi (edge colouring), dan pewarnaan wilayah (face colouring). Teknik yang dibahas pada makalah ini adalah pewarnaan simpul, teknik yang lain tidak dibahas karena teknik-teknik tersebut hanyalah bentuk lain dari pewarnaan simpul dan dapat diubah kembali menjadi model pewarnaan simpul. Misalnya pewarnaan simpul sisi adalah pewarnaan simpul dengan sisinya dianggap sebagai simpul dan pewarnaan wilayah adalah pewarnaan simpul dari graf dual planarnya.

Masalah utama dalam pewarnaan simpul graf adalah bagaimana mewarnai semua simpul pada graf sehingga tidak ada simpul-simpul yang bertetangga, yaitu dihubungkan oleh minimal satu buah sisi, memiliki warna yang sama. Hal ini biasanya kemudian dikaitkan dengan penggunaan warna yang seminimum mungkin. Jumlah warna minimum yang dapat digunakan untuk mewarnai semua simpul disebut dengan *bilangan kromatik* dari graf G, dan disimbolkan dengan *X(G)*. Contohnya graf berikut :



$\chi(G) = 4$

Gambar 1.1
Pewarnaan Simpul Graph dengan $\chi(G) = 4$

Masalah pewarnaan graf diyakini pertama kali muncul sebagai masalah pewarnaan peta, dimana warna setiap daerah pada peta yang berbatasan dibuat berlainan sehingga mudah untuk dibedakan. Hal ini kemudian mengembangkan teorema-teorema menarik dan berujung pada teorema 4 warna, yang menyatakan : *"bilangan kromatik graf planar tidak lebih dari 4."* Teorema ini pertama kali muncul sebagai suatu perkiraan oleh Francis Guthrie, seorang mantan murid dari Augustus De Morgan, pada tahun 1852 dan akhirnya dibuktikan oleh Kenneth Appel dan Wolfgang Haken. Pembuktian teorema ini menggunakan komputer dengan waktu yang melebihi 1000 jam. [7]

Masalah pewarnaan graf memiliki banyak aplikasi di dalam bidang lain, misalnya membuat jadwal, alokasi *register* pada *compiler*, penentuan frekuensi untuk radio, pencocokan pola, dan lain-lain. Masalah ini bahkan telah berkembang luas menjadi suatu permainan yang sangat terkenal di kalangan masyarakat luas, yaitu *sudoku*. Aplikasi utama yang akana dibahas pada makalah ini adalah masalah penentuan jadwal.

2. Sifat – Sifat Bilangan Kromatik Graf G

Berikut ini akan diberikan beberapa sifat graf. Sifat - sifat yang diberikan hanyalah sifat – sifat yang menyangkut bilangan kromatik (jumlah warna minimum untuk pewarnaan simpulnya), sesuai dengan tema yang dibahas pada makalah ini.

1. $\chi(G) = 1$ jika dan hanya jika G adalah graf kosong.
2. $\chi(G) \geq 3$ jika dan hanya jika G memiliki subgraph yang merupakan K3 atau C3
3. $\chi(G) \leq B(G)+1$.
4. $\chi(G) \leq B(G)$, kecuali jika G adalah graf lengkap atau graf lingkaran dengan jumlah simpul ganjil.
5. Untuk setiap graf planar, berlaku teorema 4 warna, yaitu $\chi(G) \leq 4$.
6. Bila G' adalah upagraph dari G, maka $\chi(G') \leq \chi(G)$.
7. Graf lengkap K_n memiliki $\chi(G)=n$.
8. Graf Lingkaran C_n memiliki $\chi(G)=2$ bila n genap dan $\chi(G)=3$ bila n ganjil.
9. Graf Teratur berderajat n selalu memiliki $\chi(G) \leq n+1$ sesuai sifat no 3 di atas.
10. Graf Bipartit selalu bisa diwarnai dengan 2 warna.
11. Graf yang berupa pohon selalu dapat diwarnai dengan 2 warna.

3. Aplikasi *Graph Colouring* untuk *Scheduling*

Sebagai pembukaan dari bagian ini, penulis akan memberikan sebuah contoh kasus yang menggambarkan manfaat metode pewarnaan graph dalam membuat jadwal.

Setiap semester, petugas kantor akademik MIT harus menentukan jadwal ujian. Hal ini tidaklah mudah, karena beberapa mahasiswa/i mengambil beberapa mata kuliah yang bisa bertabrakan jadwal ujiannya. Petugas tersebut ingin menghindari semua jadwal yang bertabrakan. Tentu saja, ia bisa membuat sebuah jadwal dengan setiap mata kuliah memiliki waktu ujian yang berlainan, tetapi hal ini mengakibatkan masa ujian tersebut akan sangat lama sekali dan bahkan mungkin lebih dari satu semester. Petugas tersebut tentu saja ingin membuat masa ujian sesingkat mungkin sehingga memudahkan semua pihak, bagaimana cara ia melakukannya? [3]

Di sinilah masalah pewarnaan graf memegang peranan yang penting. Kita dapat memodelkan masalah seperti di atas ke dalam model grafnya. Setiap mata kuliah dimodelkan dengan sebuah simpul pada graf tersebut. Kemudian setiap mata kuliah yang diikuti oleh setidaknya 1 orang yang sama akan memiliki hubungan yang direpresentasikan dalam bentuk sisi pada 2 simpul mata kuliah itu. Waktu atau jadwal sebuah mata kuliah berlangsung adalah warna yang diberikan pada simpul mata kuliah itu. Masalah yang harus dipecahkan berikutnya adalah bagaimana kita memberikan warna minimum kepada setiap simpul sehingga tidak ada simpul – simpul yang bertetangga memiliki warna yang sama. Hal ini merupakan masalah pewarnaan simpul graf yang telah kita singgung sebelumnya.

Contoh lain yang dapat dimodelkan dalam pewarnaan graf misalnya adalah masalah update software secara online. Contohnya pada perusahaan Akamai, versi baru dari sebuah software terus diluncurkan setiap beberapa hari ke 20.000 server yang dimilikinya. Tentu saja perusahaan tersebut tidak bisa mematikan seluruh servernya untuk mengupdatenya, ataupun mengupgrade satu per satu server tersebut karena hal tersebut akan memakan waktu yang terlalu lama. Dibutuhkan suatu cara yang efisien untuk menentukan jadwal upgrade setiap server.

Masih banyak contoh lain dari masalah pewarnaan graf, misalnya penentuan alokasi variabel pada register komputer, masalah pewarnaan peta, pemberian frekuensi radio kepada setiap stasiun radio, penentuan jadwal operasi kereta api, dan lain – lain.

Untuk masalah penentuan jadwal, masalah tersebut biasanya berhubungan dengan beberapa pekerjaan yang berhubungan seperti menggunakan sumber daya yang sama sehingga tidak bisa dilakukan pada waktu yang bersamaan.

Kita dapat menyelesaikan masalah ini dalam tiga langkah, yaitu :

1. Menggambar simpul-simpul graf
   Simpul-simpul graf yang digambarkan haruslah mewakili pekerjaan yang akan dilakukan.
2. Menggambar sisi-sisi pada graf
   Kita menggambarkan sisi-sisi pada setiap pasang simpul yang menggunakan sumber daya yang sama, yang artinya kedua pekerjaan tidak bisa dilakukan pada waktu yang sama.
3. Mewarnai graf
   Langkah terakhir yang harus kita lakukan adalah mewarnai simpul-simpul pada graf tersebut dengan warna yang minimum sehingga tidak ada simpul-simpul yang bertetangga memiliki warna yang sama.

Berikut ini diberikan sebuah contoh penyelesaian kasus mengenai penentuan jadwal.

Seorang penjaga kantor baby sitter mendapat titipan 7 orang anak, A,B,C,D,E,F,dan G.

Penjaga tersebut harus menentukan loker yang diberikan kepada setiap orangtua anak – anak tersebut sebagai tempat menyimpan peralatan mengasuhnya. Anak – anak tersebut tidak setiap waktu berada di kantor baby sitter tersebut, mereka datang dan pergi tergantung dari pesanan orang tua mereka. Karena loker di kantor tersebut terbatas dan harus digunakan sehemat mungkin, penjaga tersebut diminta untuk menentukan jadwal pemberian loker kepada setiap orang tua. Jadwal penitipan untuk setiap anak diberikan pada tabel berikut

Tabel 3.1 Jadwal Penitipan Bayi

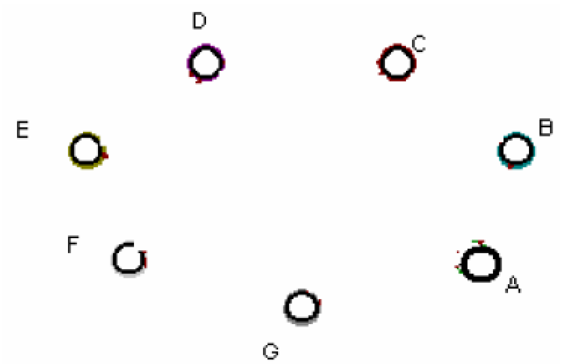
.	A	B	C	D	E	F	G
7:00	*	-	-	*	*	-	-
8:00	*	*	*	-	-	-	-
9:00	*	-	*	*	-	*	-
10:00	*	-	*	-	-	*	*
11:00	*	-	-	-	-	*	*
12:00	*	-	-	-	*	-	-

Tanda bintang (*) menggambarkan bahwa anak tersebut berada pada kantor pada waktu yang diberikan, sedangkan tanda strip (-) berarti bahwa anak tersebut tidak sedang dititipkan.

Tentukan bagaimana petugas tersebut harus menentukan loker mana yang diberikan kepada setiap orangtua. [5]

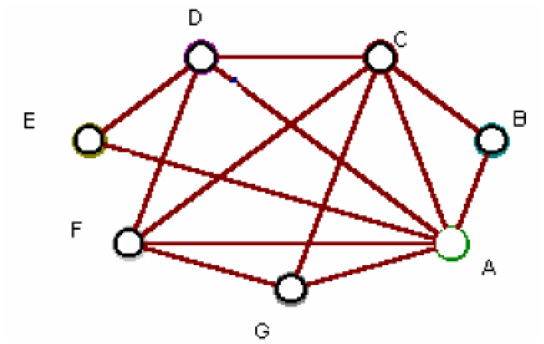
Kita akan menyelesaikan masalah di atas dalam beberapa langkah:

1. Menggambar Simpul graf
   Graf yang digambarkan di sini harus menggambarkan kondisi di atas, pertama kita membuat tujuh buah simpul.



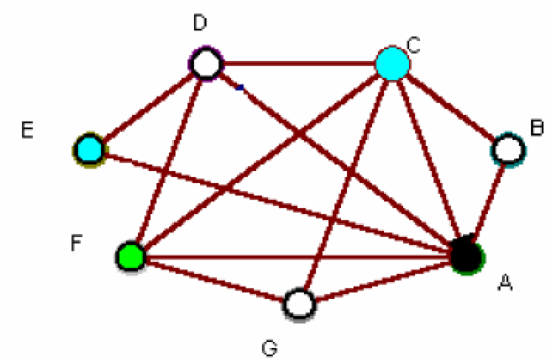
Gambar 3.1 Simpul yang Merepresentasikan Anak

2. Menggambar Sisi Graf
   Kita menggambarkan sisi - sisi pada setiap pasang simpul bila anak – anak yang dilambangkan pada kedua simpul yang dihubungkan oleh sisi tersebut menggunakan petak (slot) waktu yang sama.



Gambar 3.2 Graf yang Merepresentasikan Hubungan Jadwal Tiap Anak

3. Mewarnai Graf Tersebut
   Graf yang telah diwarnai akan terlihat sebagai berikut :



Gambar 3.3 Graf yang Telah Diwarnai

Warna minimum yang bisa digunakan untuk mewarnai graf ini adalah 4, karenanya bilangan kromatik graf ini adalah 4. Bila yang tersedia hanya tiga buah warna, maka graf ini tidak akan bisa diwarnai sesuai dengan ketentuan. Dalam hal ini, loker minimum yang tersedia harus berjumlah 4. C dan E diberikan loker 1. B, D, dan G diberikan loker 2. F diberikan loker 3. A diberikan loker 4. Pemberian loker yang sama untuk 2 orang anak atau lebih dapat dilakukan karena mereka tidak pernah berada pada tempat yang sama dalam waktu yang sama.

4. Algoritma Mewarnai Graf

Untuk graf – graf dengan sifat tertentu, kita bisa dengan muda menyebutkan bilangan kromatiknya dan mewarnainya, misalnya graf lengkap dan graf lingkaran. Tetapi umumnya graf yang kita hadapi adalah graf yang tidak memiliki keteraturan seperti itu. Karena itu, dibutuhkan algoritma yang dapat membantu kita menyelesaikan masalah pewarnaan graf.

Algoritma terbaik untuk menemukan bilangan kromatik dari suatu graf yang dikenal sampai saat ini memiliki kompleksitas waktu eksponensial untuk kasus terburuknya. Masalah ini termasuk ke dalam kelas NP Lengkap (NP Complete). Belum ada yang mampu memberikan solusi dengan kompleksitas waktu polynomial untuk kasus terburuknya, tapi juga tidak ada bukti bahwa algoritma seperti itu tidak ada. Dikatakan bahwa apabila salah satu masalah dalam kelas NP lengkap bisa diselesaikan dalam waktu polinomial, maka semua masalah di dalam kelas tersebut juga bisa diselesaikan dalam waktu polinomial. Sebaliknya bila ada bukti bahwa algoritma dengan kompleksitas algoritma polinomial tidak ada untuk salah satu masalah pada kelas NP lengkap, maka semua masalah pada kelas NP lengkap juga tidak bisa diselesaikan dalam kompleksitas waktu polinomial.

Beberapa algoritma yang telah banyak dikenal adalah: [1]

- First Fit (FF)
  Algoritma ini adalah algoritma yang termudah dan tercepat. Prinsipnya adalah mewarnai setiap simpul graf dengan warna yang tidak akan diubah lagi. Meskipun algoritma ini sangat mudah untuk diimplementasikan dan juga sangat cepat. Namun, algoritma ini memiliki probablitas besar untuk menghasilkan jumlah warna yang melebihi bilangan kromatiknya. Kompleksitas waktu asimtotik dari algoritma ini adalah O(n)

- Largest Degree Odering (LDO)
  Algoritma ini merupakan algoritma yang prinsipnya berdasarkan pada nilai derajat dari setiap simpul. Simpul yang memiliki derajat yang lebih tinggi diwarnai lebih dulu. Algoritma ini memberikan hasil yang lebih baik daripada algoritma first fit. Kompleksitas waktu asimtotik dari algoritma ini adalah O(n2).

- Saturated Degree Ordering (SDO)
  Algotima ini berprinsipkan pada jumlah warna berlainan yang ada pada tetangga-tetangga dari sebuah simpul. Simpul yang bertetanggaan dengan simpul-simpul yang memiliki lebih banyak aneka warna akan diwarnai lebih dulu. Algoritma ini memberikan hasil yang lebih baik daripada algoritma LDO. Kompleksitas waktu asimtotik dari algoritma ini adalah O(n3).

- Incident Degree Ordering (IDO)
  Algoritma ini berprinsipkan pada jumlah simpul tetangga yang telah diwarnai dari suatu simpul. Simpul yang lebih banyak bertetanggaan dengan simpul yang telah diwarnai akan diwarnai lebih dulu. Algoritma ini merupakan modifikasi dari algoritma SDO. Algoritma ini dapat dieksekusi dalam waktu yang lebih cepat, tetapi hasilnya tidak sebaik algoritma SDO. Kompleksitas waktu asimtotik dari algoritma ini adalah O(n2) Berikut ini adalah tabel yang menggambarkan jumlah warna yang dihasilkan dari setiap algortima. Kepadatan adalah perbandingan dari jumlah sisi (vertex) yang ada terhadap jumlah sisi dari graf lengkapny

Tabel 4.1 Perbandingan Efektifitas Algoritma

Jumlah Simpul	Kepadatan	FF	LDO	IDO	SDO
200	25%	20	18	18	17
200	50%	36	34	34	32
200	75%	58	55	56	53
1000	25%	64	62	63	58
1000	50%	127	123	126	116
1000	75%	217	212	214	204

5. Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari penulisan makalah ini adalah:

a. Masalah pewarnaan graf merupakan masalah yang banyak digunakan untuk memodelkan masalah di berbagai bidang, salah satunya adalah pembuatan jadwal atau scheduling.
b. Beberapa jenis graf yang memiliki keteraturan dapat ditentukan bilangan kromatiknya secara langsung dengan menggunakan sifat – sifatnya.
c. Masalah penjadwalan atau scheduling memiliki beberapa variasi yang dapat dimodelkan ke dalam masalah pewarnaan yang berbeda pula.
d. Terdapat beberapa algoritma yang dapat digunakan untuk menyelesaikan masalah pewarnaan graf. Algoritma yang cepat biasanya hasilnya tidak efisien (memberikan warna yang lebih banyak), sedangkan algoritma yang efisien biasanya membutuhkan waktu eksekusi yang lebih lama.

DAFTAR PUSTAKA

[1] Al-Omari, Hussein & Khair Eddin Sabri, New Graph Coloring Algorithms, www.scipub.org/fulltext/jms2/jms224739-741.pdf, 2006. Tanggal Akses : 2 Januari 2008, pukul 12.30 WIB.

[2] Klotz, Walter, Graph Coloring Algorithms, www.math.tuclausthal.de/Arbeitsgruppen/Diskrete-Optimierung/publications/2002/gca.ps, 2000. Tanggal Akses : 29 Desember 2007, pukul 16.00 WIB.

[3] Leighton, Tom & Ronitt Rubinfeld, Graph Teory, http://theory.lcs.mit.edu/classes/6.042/fall06/lec6.pdf, 2006. Tanggal Akses : 29 Desember 2007, pukul 15.00 WIB.

[4] Marx, Daniel, Graph Coloring Problems And Their Applications In Scheduling, http://citeseer.ist.psu.edu/672702.html, 2006. Tanggal Akses : 29 Desember 2007, pukul 14.30 WIB.

[5] Mawata, Christopher, Graph Teory Lesson, http://oneweb.utc.edu/~Christopher-Mawata/petersen/answers/les8Ans.html, 2006. Tanggal Akses : 29 Desember 2006, pukul 14.15 WIB.

[6] Munir, Rinaldi, Diktat Kuliah IF 2153, Matematika Diskrit, Edisi Keempat, Program Studi Teknik Informatika, STEI, ITB, 2006.

[7] Rosen, Kenneth H., Descrete Mathematics and Its Applications, 4 th, McGraw-Hill Internasional, 1999.

[8] The Computer Action Team, Graph Colouring Algorithm, http://web.cecs.pdx.edu/~postj/graph/graph.html, 2005. Tanggal Akses : 29 Desember 2007, pukul 15.30 WIB.

[9] Wikipedia, Wikipedia – Free Encyclopedia, www.wikipedia.com, 2006. Tanggal akses : 27 Desember 2007, pukul 15.00 WIB.